数理解析研究所講究録 1426 巻 2005 年 263-269

パラメタ付き時間インターバルオートマトンに対するパラメトリック検証の高速化手法

大阪大学・大学院情報科学研究科　橋本 英明 (Hideaki HASHIMOTO)
谷本 匡亮 (Tadaaki TANIMOTO)
中田 明夫 (Akio NAKATA)
東野 輝夫 (Teruo HIGASHINO)
Graduate School of Information Science and Technology,
Osaka University

1 はじめに

近年，システム検証技術の重要性が増している．代表的な検証技術としてモデル検査が知られている．モデル検査はシステムを状態遷移モデルで記述し，そのモデルが安全性や活性などの望ましい性質を満足するか否かを検証する技術である．現在，さまざまなモデル検査ツールが存在しているが，遅延やタイムアウトなど，時間に関してシステムが持つさまざまな設計パラメタを持つ実時間システムに対するモデル検査においては，モデルが与えられた性質を満たすための設計パラメタに関する条件 (パラメタ条件) を自動導出し，設計者はパラメタ条件を満たす範囲内で設計パラメタを自由に選べるようにすることが望ましい．このようなパラメタ条件を導出する手法はパラメトリックモデル検査手法 [5] と呼ばれている．既存のパラメトリックモデル検査ツールとしては HyTech[4],TReX[2] などがある．これらによりパラメタを用いて実時間システムモデルを記述し，与えられた性質を満たすためのパラメタ条件を自動導出することが可能となったが，計算量が大きく状態数やパラメタ数が大きいモデルを扱えないなどの問題点がある．

この問題の部分的な解決法としては，システムのモデルをある部分クラスに限定し，そのクラスに特化したより効率のよいパラメトリックモデル検査アルゴリズムを開発することが考えられる [6]．一方，Web サービス仕様などを対象とした検証問題 [3] の場合，システムの動作自体は時間に依存しないことが多い．そのようなシステム全体の性能を保証するため，全体の処理に時間制約を課して個々の動作に対する時間制約を求めることができれば，動作の正しさのみならずシステム全体の性能をも保証することができる．この場合，時間制約の記述能力としては，個々の動作 (遷移) の処理時間に対する制約を一般にパラメタ変数を用いて記述された時間インターバルの形式で与えることができれば十分である場合が多い．

そこで本研究では対象とする部分クラスとしてパラメタ付き時間インターバルオートマトン (Parametric Time-Interval Automata:PTIA) を定義し，PTIA を対象としてパラメトリックモデル検査を高速化する手法を提案する．PTIA はクロック変数を 1 つだけ持ち，現状態から次状態に遷移するまでに許される経過時間の上限と下限を，一般にパラメタ変数を含む式で指定するような時間オートマトン [1] の部分クラスである．

本論文では PTIA を対象としたパラメトリックモデル検査の高速化に関する以下の 2 点の手法を提案する．

1. 単一の PTIA に対する、性質を保存した状態削減によるパラメトリックモデル検査高速化手法

2. 複数の PTIA の積オートマトンが PTIA のクラスに入るか否かの判定および単一の PTIA への変換手法

上記 1. の手法の概要は次の通りである．検証性質に関連して着目する動作を外部動作，それ以外の動作を内部動作とみなし，検証者は PTIA による動作記述，および外部動作の指定を入力として与える．次に，有限オートマトンから正規表現 (regular expression) を導出するアルゴリズム [7] を用いて内部動作を縮約する．その際，付加されている時間制約は，オートマトン理論における正規表現上の演算に帰着して保存する．このようにして変換した PTIA は，元の PTIA の外部動作に対する実行可能系列の集合と，その実行時刻を保存

したものとなる．

上記 2. の手法の概要は次の通りである．一般に，n 個の並行モジュールが協調して 1 つの処理を行うような並行システム (n_Sys) を検証する場合，n 個の各モジュールの動作がそれぞれ PTIA で記述されているとしても，n_Sys 全体の動作 (n 個の PTIA の積オートマトン) を記述するためには一般にクロック変数が n 個必要である．しかし，もし n_Sys の全体動作が 1 つのクロック変数を用いた PTIA で記述可能であることが分かれば，上記 1. の高速化手法に帰着することができる．

本論文の以降の構成は次のとおりである．まず，2 章で PTIA の形式的な定義を行う．3 章では単一 PTIA に対する状態削減手法について説明する．4 章では複数 PTIA の積が単一 PTIA に変換できるための十分条件およびそれを利用したパラメトリックモデル検査高速化手法について説明する．5 章では例題に対する実験結果を示す．6 章で結論と今後の課題を述べる．

2 パラメタ付き時間インターバルオートマトン

動作の集合を Act，パラメタ変数を含む変数の集合を Var とする．$\boldsymbol{R}$ を実数全体の集合，$\boldsymbol{R}^+$ を非負実数全体の集合とする．$Intvl(Var)$ を $e_1 \leq t$，$\leq t \leq e_2$，または $e_1 \leq t \wedge t \leq e_2$ のいずれかで表される式の集合と定義する．ここで，e_1 と e_2 は $Var \setminus \{t\}$ 上の変数と R 上の定数からなる線形式 (加減算のみ用いた式) とする．

定義 1 (パラメタ付き時間インターバルオートマトン (PTIA)) とは 5 字組 $< S, t, PVar, E, s_{init} >$ である．ただし，S は状態の有限集合，$t \subseteq Var$ はクロック変数，$PVar \subseteq Var$ はパラメタ変数の有限集合，$E \subseteq S \times (Act \cup \{\tau\}) \times Intvl(PVar) \times S$ は遷移関係の有限集合，$s_{init} \in S$ は初期状態である．ここで，τ は内部動作を表す．一方，Act 上にある他の全ての変数は観測可能な動作として表現される． □

クロック変数 t とパラメタが条件式 P を満たすとき，状態 s_i から動作 a が実行可能となり，その後状態 s_j に遷移することを $s \overset{a@?[P]}{\rightarrow} s'$ と略記する．状態 s_j に遷移した後，クロック変数 t は 0 にリセットされる．

PTIA のセマンティクスは一般の時間オートマトンと同様である．なお，クロックとパラメタの値は，写像 $\sigma : Var \mapsto \mathbf{R}$ で与えられる．これを代入と呼ぶ．このような代入全体の集合を Val で表す．式 $P \in Intvl(Var)$ に対して代入 σ を適用した式が真であることを $\sigma \models P$ と記述する．時間オートマトンのセマンティクスは，状態 s と変数への代入 σ の組 (s, σ) を状態とするラベル付遷移システムとして定義される．この組を具体的状態と呼ぶ．$CS \overset{\text{def}}{=} \{(s, \sigma) | s \in S, \sigma \in Val\}$ を具体的状態の集合とする．

3 単一 PTIA に対するパラメトリックモデル検査高速化

本章では単一の PTIA に対するパラメトリックモデル検査高速化手法について述べる．

まず，PTIA の各動作に対して，検証性質に関する外部観測性の概念を導入する．具体的には，検証性質として例えば「指定した動作 a から動作 b まで c 単位時間以内に実行可能」という性質が与えられたときには，動作 $a, b \in Act$ を外部から観測可能な動作 (外部動作)，それ以外の動作を内部動作 τ とみなす．

状態削減の際に上記のような性質を保存するためには，直観的には「内部動作を観測できないものとし，実行可能な外部動作の系列 (トレース) の集合が，外部動作の実行時刻も含めて等しい」という性質を保存すれば十分である．より形式的には以下のように定義される．

定義 2 (時間弱遷移関係) 時間 LTS$< CS, Act \cup \mathbf{R}^+ \cup \{\tau\}, CE, (s_{init}, \sigma_{init}[t \rightarrow 0]) >$ の状態 (具体的状態) 集合 CS 上の 2 項関係として，**時間弱遷移関係** $\rightarrow_{\mathrm{w}}$ を次のように定義する：

1. $(s,\sigma) \xrightarrow{\tau}_w (s',\sigma') \stackrel{\text{def}}{=} \exists k \in \mathbf{N_0}$ s.t. $(s,\sigma)(\xrightarrow{\tau})^k(s',\sigma')$
   ただし，$\mathbf{N_0}$ は 0 以上の整数 (自然数) の集合を表し，集合 CS 上の 2 項関係 R に対して R^k は，$R^0 \stackrel{\text{def}}{=} \{((s,\sigma),(s,\sigma))|(s,\sigma) \in CS\}$，$R^k \stackrel{\text{def}}{=} R \cdot R^{(k-1)}$ と帰納的に定義される 2 項関係であるとする．
2. $(s,\sigma) \xrightarrow{v}_w (s',\sigma')$ $(v \in \mathbf{R}^+)$
   $\stackrel{\text{def}}{=} \exists v_1, v_2, ..., v_n \in \mathbf{R}^+ \ [\, v = \sum_{i=1}^n v_i$
   $\wedge \exists s_1, \sigma_1, \sigma_1', s_2, \sigma_2, \sigma_2', ..., s_n, \sigma_n, \sigma_n'$
   s.t. $(s,\sigma) \xrightarrow{\tau}_w (s_1,\sigma_1) \xrightarrow{v_1} (s_1,\sigma_1') \cdots \xrightarrow{\tau}_w (s_n,\sigma_n) \xrightarrow{v_n} (s_n,\sigma_n') \xrightarrow{\tau}_w (s',\sigma')\,]$
3. $\xrightarrow{a}_w$ $(a \in Act) \stackrel{\text{def}}{=} \xrightarrow{\tau} \xrightarrow{a} \xrightarrow{\tau}_w$ □

定義 3 (時間弱トレース集合) PTIA $M = < S, \{t\}, PVar, E, s_{init} >$ と代入 σ に対して，M が実行可能な外部動作および実行時間の系列の集合 $\mathcal{L}_M(\sigma)$ を以下のように定義し，M の σ に関する時間弱トレース集合と呼ぶ：

$$\mathcal{L}_M(\sigma) \stackrel{\text{def}}{=} \{(v_1,a_1)\cdots(v_n,a_n)|\exists s_1,\ldots,s_n \in Act, v_1,\ldots,v_n \in \mathbf{R}^+, a_1,\ldots,a_n \in Act,$$
$$[(s_{init},\sigma) \xrightarrow{v_1}_w \xrightarrow{a_1}_w (s_1,\sigma)\cdots(s_{n-1},\sigma) \xrightarrow{v_n}_w \xrightarrow{a_n}_w (s_n,\sigma)]\ \}$$

□

定義 4 ((パラメトリック) 時間弱トレース等価性) PTIA $M_1 = < S_1, \{t_1\}, PVar_1, E_1, s_{1init} >$ および $M_2 = < S_2, \{t_2\}, PVar_2, E_2, s_{2init} >$ が (パラメトリック) 時間弱トレース等価であるとは，パラメタ変数への任意の代入 σ に対して $\mathcal{L}_{M_1}(\sigma) = \mathcal{L}_{M_2}(\sigma)$ が成立することであると定義する． □

状態削減の際，各遷移の時間制約は正規表現の演算 [7] に帰着させる．削減前と削減後の PTIA が，内部動作を ε-遷移とみなしたときに外部実行可能な系列の集合が実行可能な時刻も含めて等しいこと (時間弱トレース等価性) を保存することを示す．

3.1 正規演算 (連接) への帰着

連接への帰着は図 1 のように実現できる．図 1 の上半分は遷移ラベルに正規表現が記述可能な，一般化非決定性有限オートマトン (GNFA)[7] である．GNFA では状態 S_0 から R1 で状態 S_1 へ遷移し，状態 S_1 から R2 で状態 S_2 へ遷移する．このとき GNFA では正規表現の連接演算を用いて状態 S_0 から R1·R2 で状態 S_2 へ遷移するように等価変換し，状態を縮約することができる．これに対応して，図 1 の下半分の PTIA では状態 S_0' から状態 S_1' への遷移にかかる時間制約と，状態 S_1' から状態 S_2' への遷移にかかる時間制約の連接演算を定義することにより，同様の縮約が可能となる．形式的には以下の命題が成立する．

命題 1 PTIA M が動作系列 $s_1 \xrightarrow{\tau@?[P_1(t)]} s_2 \xrightarrow{\alpha@?[P_2(t)]} s_3$ $(\alpha \in Act \cup \{\tau\})$ を持ち，状態 s_2 に入る遷移および出る遷移が他に存在しないならば，M の状態 s_2 を削除し，新しい遷移 $s_1 \xrightarrow{\alpha@?[(P_1 \cdot P_2)(t)]} s_3$ を加えて得られる PTIA M' は M と時間弱トレース等価である．ただし，$P_1 \cdot P_2$ は以下のように定義される $Intvl(PVar)$ 上の 2 項演算とする：

$$(P_1 \cdot P_2)(t) \stackrel{\text{def}}{=} \exists t_1 \exists t_2 [P_1(t_1) \wedge P_2(t_2) \wedge t = t_1 + t_2]$$

□

3.2 正規演算 (閉包) への帰着

閉包への帰着は図 2 に示すように実現できる．前節と同様，正規表現の $*$-閉包に対応する時間インターバル上の演算を定義することにより，GNFA 上と同様の縮約が可能となる．ただし，検証性質を満たすための

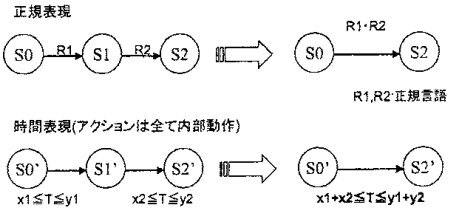

図 1: 連接

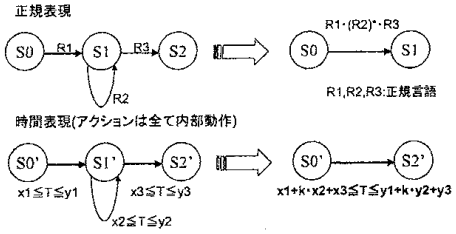

図 2: 閉包

ループ回数に関する条件を求める目的のために，ループ回数をパラメタ化する．形式的には以下の命題が成立する．

命題 2 PTIA M が動作系列 $s_1 \xrightarrow{\tau@?[P_1(t)]} s_2 \xrightarrow{\tau@?[P_3(t)]} s_2 \xrightarrow{\alpha@?[P_3(t)]} s_3$ $(\alpha \in Act \cup \{\tau\})$ を持ち，状態 s_2 に入る遷移および出る遷移が他に存在しないならば，M の状態 s_2 を削除し，新しい遷移 $s_1 \xrightarrow{\alpha@?[(P_1 \cdot \mathcal{L}_k(P_2) \cdot P_3)(t)]} s_3$ を加えて得られる PTIA を $M'(k)$ とするただし，$\mathcal{L}_k(P)$ は新しいパラメタ変数 k を自由変数として含み，以下のように定義される $Intvl(PVar)$ 上の単項演算とする：

$$\mathcal{L}_k(P)(t) \stackrel{\text{def}}{=} \exists t'[P(t') \wedge t = k \cdot t']$$

このとき，$M'(k)$ と M は以下の意味で時間弱トレース等価である：

$$\forall \sigma[\mathcal{L}_M(\sigma) = \bigcup_{k \in \mathbf{N_0}} \mathcal{L}_{M'(k)}(\sigma)]$$

ただし，$\mathbf{N_0}$ は 0 以上の整数全体の集合とする． □

3.3 一般の場合

前節までの議論を一般化すると次の定理が成り立つ：

定理 1 (単一 PTIA に対する時間弱トレース等価性を保存した状態数削減) PTIA M の遷移の集合 IN,OUT,$LOOP$ をそれぞれ $IN \stackrel{\text{def}}{=} \{s^i_{in} \xrightarrow{\tau@?[P^i_{in}]} s_{rip} \mid i \in I, s^i_{in} \neq s_{rip}\}$, $OUT \stackrel{\text{def}}{=} \{s_{rip} \xrightarrow{\alpha_j@?[P^j_{out}]} s^j_{out} \mid j \in J, \alpha_j \in Act \cup \{\tau\}, s^j_{out} \neq s_{rip}\}$, $LOOP \stackrel{\text{def}}{=} \{s_{rip} \xrightarrow{\tau@?[P^l_{loop}]} s_{rip} \mid l \in L\}$ とする．このとき，M から状態 s_{rip} を除去し，全ての $i \in I$, $j \in J$, $l \in L$ の組み合わせに対して新しい遷移 $s^i_{in} \xrightarrow{\alpha_j@?[P'_{in} \cdot \mathcal{L}_{k_l}(P^l_{loop}) \cdot P^j_{out}]} s^j_{out}$ を追加して得られる PTIA M' は，新しく導入されたループパラメタ群 $\{k_l | l \in L\}$ を持ち，M より状態数が 1 少なく M と以下の意味で時間弱トレース等価な PTIA である：

$$\forall \sigma[\mathcal{L}_M(\sigma) = \bigcup_{l \in L} \bigcup_{k_l \in \mathbf{N_0}} \mathcal{L}_{M'}(\sigma)]$$

(証明のアイデア) 文献 [7] に示されている，有限オートマトンから GNFA に変換し，GNFA を縮約することにより正規表現を求めるアルゴリズムの正当性証明と同様に示すことができる．詳細は省略する． □

与えられた PTIA に対して，定理 1 で示した操作をこれ以上適用できなくなるまで繰り返し適用することにより，元の PTIA と時間弱トレース等価で，かつ，状態数がより小さい PTIA を得ることができる．

4 並列動作するPTIA群に対するパラメトリックモデル検査高速化

PTIA群の積オートマトンは通常の時間オートマトンの積[1]と同様に定義する．ただし，各PTIA同士で行う通信は，他のPTIAも同じ動作名の動作を持つ場合はその動作を持つ全てのPTIAが実行可能になるまで待って同期実行し，それ以外の動作は各PTIA独立に非同期で実行するものとする．

以降ではPTIA群の積オートマトンを時間弱トレース等価な単一PTIAへ変換できるための条件の定義，および，変換法を示す．まず，準備として必要な概念を以下に定義する．

定義 5 (遷移の非同期並列/同期並列関係) PTIA群 $M_1 =< S_1, t, PVar_1, E_1, s^1_{init} >, \ldots, M_n =< S_n, t, PVar_n, E_n, s^n_{init} >$ の任意の積状態 $(s_1, \ldots, s_n) \in S_1 \times \cdots \times S_n$ および $\{s_1, \ldots, s_n\}$ の部分集合 $\{s_{i_1}, \ldots, s_{i_k}\}(2 \leq k \leq n)$ に対して，遷移の集合 $\{s_{i_1} \xrightarrow{a_{i_1}@?[P_{i_1}]} s'_{i_1}, \ldots, s_{i_k} \xrightarrow{a_{i_k}@?[P_{i_k}]} s'_{i_k}\}$ を，動作名 $a_{i_1}, \ldots, a_{i_k}$ が互いに異なるならば非同期並列であると呼び，動作名 $a_{i_1}, \ldots, a_{i_k}$ が全て等しいならば同期並列であると呼ぶ． □

非同期並列/同期並列の概念を用いて，PTIA群の積オートマトンが単一PTIAに変換できるか否かに関して以下の定理が成り立つ．

定理 2 (並行 PTIA 群から単一 PTIA への変換可能性条件) PTIA群 $M_1 =< S_1, t, PVar_1, E_1, s^1_{init} >, \ldots, M_n =< S_n, t, PVar_n, E_n, s^n_{init} >$ は，以下の条件を満たすとき $M_1, \ldots, M_n$ の積オートマトンと時間弱トレース等価な単一PTIAが存在する：

1. 任意の積状態 $(s_1, \ldots, s_n) \in S_1 \times \cdots \times S_n$ に対する任意の非同期並列な遷移集合を $Async = \{s_{i_1} \xrightarrow{a_{i_1}@?[P_{i_1}]} s'_{i_1}, \ldots, s_{i_k} \xrightarrow{a_{i_k}@?[P_{i_k}]} s'_{i_k}\}$ としたとき，$M_1, \ldots, M_n$ の積オートマトンの具体的状態 $((s_1, \ldots, s_n), \sigma)$ からのある実行系列において，もし動作 a_{i_l} が他のある動作 $a_{i_j}(j \in \{1, \ldots, k\} \setminus \{l\})$ の後に出現するならば，動作 a_{i_l} の時間制約 P_{i_l} は $P_{i_l} = [true]$ である．
2. 任意の積状態 $(s_1, \ldots, s_n) \in S_1 \times \cdots \times S_n$ に対する任意の同期並列な遷移集合を $Sync = \{s_{i_1} \xrightarrow{b@?[P_{i_1}]} s'_{i_1} a_1, \ldots, s_{i_k} \xrightarrow{b@?[P_{i_k}]} s'_{i_k} a_k\}$ としたとき，もし $P_{i_l} \neq [true]$ であるような時間制約 P_{i_l} が存在するならば，l 以外の他の全ての遷移の時間制約 P_{i_j} $(j \in \{1, \ldots, k\} \setminus \{l\})$ は全て $P_{i_j} = [true]$ である． □

紙面の制限のため，証明の詳細は省略する．

5 実験

本論文では，適応可能な検証対象の一つとして，卸売業における商品調達システムへの応用について述べる．

5.1 実験の概略仕様

卸売業における商品調達システムの概略仕様を以下に示す．全体のシステム構成は図3のようになる．

図3において，各システム間の送信遅延を $d1 \leq t \leq d2$ のようにパラメタを用いて表現する．送信動作は以下のように対応する．

- request,finish:卸売業への注文要求，注文終了要求
- resist1,resist2:取引先への在庫確認，発注要求
- search:レジストラへの取引先検索要求

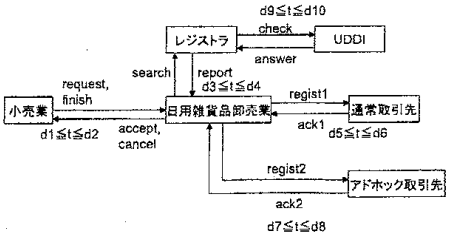

図 3: 卸売業における商品調達システム

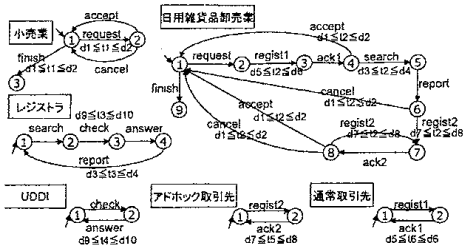

図 4: 各システムの PTIA

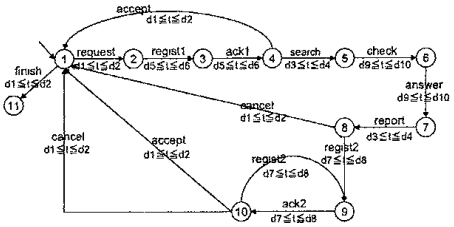

図 5: 図 4 の並列合成と時間弱トレース等価な単一 PTIA

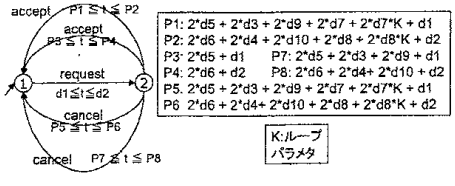

図 6: 図 5 の状態数を削減した単一 PTIA

- check:UDDI への情報取得要求
- report:卸売業への結果報告
- answer:レジストラへの情報照会結果報告
- ack1,ack2:卸売業への在庫照会結果，受注報告

図 3 と各システムの構成から，各システムの PTIA を求めると図 4 のようになる．図 4 において時間の記述がなされていない遷移は全て時間制約 $[true]$ で遷移するものとする．

検証性質としては次のものを考える．

「小売業が商品の注文を出してから，日用雑貨品卸売業が結果を返すまでの処理を 10 秒以内に完了したい．」

5.2 提案手法によるパラメタ条件導出

提案手法では，まず図 4 の PTIA 群の積オートマトンが弱時間トレース等価な単一 PTIA に変換可能かどうかの判定を行った．具体的には，PTIA 群の積オートマトンの初期状態から定理 2 の条件を満たしているか判定していく (判定の経過については紙面の都合により割愛する)．条件判定を続けていくことで図 5 のような，時間弱トレース等価な単一 PTIA に変換することができる．

次に図 5 の PTIA に対して，3 章で述べた高速化を実行する．検証性質より，request,accept,cancel を外部動作とし，状態数を削減する．実行結果は図 6 のようになる．

図 6 より，この後既存のツールを使用することなく，検証性質からパラメタの条件式が，$P2 < 10$ *and* $P4 < 10$ *and* $P6 < 10$ *and* $P8 < 10$ であることが導出可能である．

5.3 比較

表 1 に実行結果を示す．実験環境は，CPU:Pen4 2.8GHz, Memory:1GB である．

表 1: 実行結果

使用したツール	対象とするクラス	計測結果
TReX	パラメタが記述可能な時間オートマトン	処理が終了しない (3 日以上))
UPPAAL	パラメタが記述不可能な時間オートマトン	0.03～0.06
提案手法	パラメタの記述範囲を制限した時間オートマトン (PTIA)	0.04+α

ここで，提案手法の計測結果 0.04+α のうち，0.04 は状態数削減にかかった時間である．PTIA 群の積オートマトンが時間弱トレース等価な単一 PTIA に変換可能かどうかの判定は手動で行った．その所要時間は数 10 分程度であった．

TReX は対象とするクラスの範囲が広く，さまざまな例を扱うことが可能である．しかし，パラメタの場合分けなどで処理時間が大きくなっている．提案手法では，記述可能なクラスを制限することにより，パラメタを扱うことができない UPPAAL と同等の処理時間で結果を得ることができた．

6 あとがき

本研究では，パラメタ付き時間インターバルオートマトン (Parametric Time-Interval Automata:PTIA) の定義と，PTIA を対象としてパラメトリックモデル検査手法を高速化する手法の提案とその実装を行った．

提案した手法を実験例題に適応した結果，現実的な時間でパラメタの条件式を導出することができた．

今後の課題としては，様々な例題，様々なシナリオに適応し，本手法の有効性を確認することが挙げられる．

参考文献

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] A. Annichini, A. Bouajjani, and M. Sighireanu. TReX: A tool for reachability analysis of complex systems. In *Proc. of the 13th Int. Conf. Computer Aided Verification(CAV 2001)*, Vol. 2102 of *Lecture Notes in Computer Science*, pp. 368–372. Springer, June 2001.

[3] X. Fu, T. Bultan, and J. Su. Analysis of interacting BPEL web services. In *Proc. of the 13th World Wide Web Conference(WWW 2004)*, p. 621, May 2004.

[4] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Int. Journal on Software Tools for Technology Transfer*, 1(1-1):110–122, 1997.

[5] P. Matousek. Tools for parametric verification. a comparison on a case study. In *Proc. of IEEE TC-ECBS and IFIP WG10.1 Joint Workshop on Formal Specifications of Computer-Based Systems (FSCBS2004)*, pp. 45–55, 2004.

[6] A. Nakata and T. Higashino. Deriving parameter conditions for periodic timed automata satisfying real-time temporal logic formulas. In *Proc. of 21st IFIP TC6/WG6.1 Int'l Conf. on Formal Techniques for Networked and Distributed Systems (FORTE2001)*, pp. 151–166. Kluwer Academic Publishers, Aug. 2001.

[7] M. Sipser. *Introduction to the Theory of Computation*, chapter 1. PWS Publishing Company, 1st edition, 1996.